Choosing a Linux OS for Desktop Use

An in-depth guide by Space Banana & Pepi

Are you a new or experienced user with Unix-like operating systems, such as systems of the Linux family? Maybe you are looking for a new system to use on a daily basis, but with the huge amount of choice there is out there, it can be very hard and confusing to settle on something for once and for all. This guide will help you choose what system of the Linux family to use for daily desktop use.

What is a "distro"?

A distro, or distribution, is commonly used to refer to an operating system of the Linux family that uses the mainline Linux kernel, alongside a set of userspace software and its own unique software, such as the package manager. These systems are called distributions because they are a way to "distribute" the Linux kernel to real world use in an operating system that can be used by a human.

These systems are usually called "Linux" as an abbreviation, as if Linux was the operating system. The truth is that Linux not an operating system, it's just a kernel, the core of the operating system. Linux communicates with your computer's hardware, manages memory, manipulates the disk, filesystems, partitions, etc. All operating systems need a kernel, and the kernel contains a set of system calls for software to use. When a software is compiled for "Windows" or "Linux", it is actually compiled to use system calls of the Windows and Linux kernels. The same applies to other operating systems and their kernels.

So, if Linux is not an operating system, then why are all these systems abbreviated as "Linux"? Linux-based systems, or distros, use the mainline Linux kernel, with just a few patches and firmware added on top. Software for "Linux" will then in theory work on all of these operating systems, assuming anything else it depends on is also satisfied. Linux distros are therefore intercompatible.

The structure of the OS

Systems of the Linux family have their own unique traits, but also many common similarities. A minimal but functional Linux operating system is usually composed by these pieces:

- Bootloader
- Kernel + drivers/firmware
- Init system
- Shell
- CLI tools

The bootloader is simply a piece of software that is self-sufficient, not tied to any OS, and lets the user boot to an operating system.

After booting to your Linux system, the kernel will be initialized and immediately handle memory, hardware, and much more. As said earlier, the kernel is the core of the operating system, and you cannot have an operating system without a kernel.

All software after the kernel in hierarchy is called "userspace". After the kernel is initialized, your userspace is launched. The kernel will launch your init system, which is responsible for handling your system services, among other similar tasks.

After the init system is initialized, you will now be able to log in to your system and interact with it. At its very minimal core, raw system interaction would imply raw commands, composed of the path to a program and its command-line arguments. The user cannot do this without the help of a shell. A shell provides a command-line interface so you can prompt your commands. Shells also come with their own syntax and text interpretation/processing so you can write elaborate and powerful commands. Through the shell, the user can now interact with the system and execute software. You now have a functional system.

Of course you still need actual software to do anything, that's why the system is bundled with core software, such as kernel utilities, ncurses functions like "clear" and a set of core administration utilities, bundled with crucial commands such as "ls", "cd", "uname", etc.

You can then also have bigger and more complex software to have audio in your system as well as a graphical interface.

Linux-based systems also come with a package manager, which will be discussed later.

Differences between distros

Generally, a distro is only differenciated by these characteristics:

- Kernel patches and extra firmware/drivers
- Default bootloader
- Init system
- Core utilities
- C standard library
- Default shell
- Package manager
- Package repository
- Release cycles
- Installer

- Default setup and preinstalled software
- Unique software made by the developers

As you see, the difference between distros doesn't matter much on the surface-level, such as what desktop environment it uses, or how it looks like, or if it comes with an office suite by default. What matters the most is the core of the system and its functioning.
Different systems will come with different kernel patches and firmware, although the difference isn't usually noticeable.

The default bootloader can also be changed in most cases. Usually you either choose GRUB or SystemD-boot.

The init system cannot usually be changed. Most distros will ship with SystemD, others will ship with OpenRC, others will ship with Runit. A few of them might ship with more than 1 supported init system. If you are concerned about the init system, then look out for distros that have what you need. However, if you don't care or are not really sure why you should use a different init system, then you will be fine with the most popular, SystemD.

Most distros ship with the GNU coreutils, although some systems come with Busybox, and rarely BSD utils. You usually cannot change the core system utils of your distro, so this is an important factor when choosing your system.

Another component you cannot replace is the C standard library that comes with the system. Most systems will ship the GNU C library, although some systems come with the Musl C library. If you run standalone third-party software, such as software that you find on its official website or Git repo, then most likely the binary executable is linked to the GNU C library. If your system uses Musl instead, you will have to do some workarounds.

In most systems, the default shell is Bash, although it's pretty simple to replace with any other shell, such as Zsh, Dash, Ash, Fish, etc.

The package manager and its repository are among the most important differences in a distro. The package manager is responsible for installing, managing and updating the software from your system's native repository. The package manager also shares and controls dependencies, and cleans up unused old packages. It also controls the versions of the packages, as well as package conflicts. While most package managers do a good job, you should still have a concern over the package repository.

Your package repository will tell you how many packages it has, as well as its versions and variants. You generally want a big repository.

Some package managers, like Gentoo and NixOS's package managers, are source-based, meaning they can grab a package's source from the repository and easily compile the software.

The packaging release cycle is highly important, and probably the big reason to pick one distro over another. The package cycle tells you how your system updates, as well as how frequently. A stable release (or point release) distro will

have major updates between versions, such as Debian updating from 11 to 12. Some of these stable release distros have small updates in-between, such as Fedora updating software that isn't critical to the system, while leaving the bigger and less safe updates to the next major version. How often it updates also matters. NixOS's stable channel updates twice a year, while Debian's stable channel updates every 2 years.
A rolling release distro will update packages as soon as they are deemed ready. These distros don't have a versioning system like stable release distros, since they update as packages are ready to be launched to the general public. These systems have more up-to-date software in the repository, at the cost of being less predictable, as they might push software with bugs, or handle packages incorrectly occasionally.

The installer also matters. Some installers are GUI-based, others TUI-based, others are fully manual procedures. Some installers are simpler than others. Some installers let you customize your system more than others. All of this matters.

The default system setup also matters. Distros like Linux Mint and Ubuntu will come with lots of software, alongside their desktop, while systems like Arch, NixOS, etc let you set up a minimal system with little software out-of-the-box.

Lastly, some distros come with their unique software, such as OpenSUSE's YaST, or Manjaro's GUI software manager.

Choosing a Distro

Now that you know what to expect and look for in a distro, we recompiled a list with advantages and disadvantages of all the main distros in each category, and a small overview of what the category means in practice.
Remember to check every category, as a distro might shine more in one area than other.
The intended way to read this is to check every category in order, things said at the beginning may also apply to other distros in the list. You can skip categories, if you are interested in, for example, a minimal distro, just go straight to that section.

Stable release systems

This category covers the distros that have a defined release cycle in place.

- **Debian Stable:**

  One of the oldest and most battle-tested distros. It uses a “full-freeze” method to control, manage and distribute software.
  What this means is not updating any piece of software inside the repositories outside of security fixes or the occasional bug fix.

  Debian’s biggest advantage is its release schedule. The distro developers

follow a rigorous series of steps that squash out as many bugs and problems as humanly possible, allowing for very reliable systems worthy of being used in critical environments like government infrastructure.

You can read a detailed explanation at:
https://release.debian.org/trixie/freeze_policy.html

This is a double edged sword, you might need more recent software for your work. Workarounds do exist, like using third party package managers such as Flatpak, Snap, Nix, etc.
Allowing a stable system with a few frequently updated packages.

Debian has its fair share of derivatives. The usual thing is that they will follow Debian's upgrading path, but you can see them doing things differently, like taking more time as they polish their user experience for the new Debian release.
Follow the derivative's news for when and how to update, as they do modifications and very rarely are they fully compatible with Debian.

- **Fedora:**

  Fast and slow at the same time. Fedora balances on a rope, as it offers a mix of a rolling release with characteristics of a stable release.

  It follows a 6 month release schedule, with support given up to 13 months, allowing you to skip a version, such as upgrading directly from release 30 to 32.
  Fedora's method to deliver software is to freeze critical system packages that need to be stable, or at least delay new versions of the software until they feel comfortable releasing it to their users.
  Meanwhile the rest of the software updates at a fast pace, like what you would see in a rolling release, although slightly slower.
  Big or critical package updates are released between major versions, while smaller and less critical/unsafe package updates are pushed regularly.
  It's usual to see the releases getting delayed, as the Fedora team wants to ensure a problem free launch.

  Fedora's selling point is adopting new technologies quickly, while providing an experience as stable as possible.
  Another double edged sword, as you can feel used as a test subject due to the speed of adoption.

- **OpenSUSE Leap**

  SUSE Enterprise made free. The OpenSUSE team uses open source to its fullest, replicating the paid enterprise distribution's publicly available code on a free package, of course without the technical support that SUSE bundles with their product.
  Thankfully bug fixes and security updates are not SUSE Enterprise

exclusive.

Leap uses the following method for software distribution.
Every 36-48 months, pretty soon after SUSE's release, a mayor release that updates every package in the system to a recent version is distributed.
After that, once per year, point releases that upgrade some of the software, like desktop environments are distributed. But things like the kernel stay at older versions, with some critical backports introduced.

This distro shares the same disadvantages as Debian Stable.

Leap's future as a distro is currently uncertain, because SUSE is working on ALP, an immutable approach that will replace SUSE Enterprise. Leap 15.6 is going to be released, so the window to change distro is not critical, this is a developing story.

OpenSUSE is developing a new distro that's closer to Leap.

- **OpenSUSE Slowroll**

  This is the new offering currently in the works.
  It's an alternative for Tumbleweed users that find a rolling release model too aggressive, but a slower 6 month model too slow.

  Once every 1-2 months, a new version that upgrades everything is released. Bug fixes and security updates are consistently released no matter the freezing status of Slowroll.

- **NixOS**

  A niche that needs time.
  NixOS offers a completely different method of distributing and managing software, based on the Nix package manager.
  This is an extremely powerful approach, as it allows for reproducibility, very hard to break and a full declarative configuration. You can configure your whole NixOS system, your package setup and your system-wide configuration files through the configuration.nix file, written in the Nix programming language.

  NixOS is also immune to dependency-hell and has a huge repository, composed of more than 80000 packages.

  NixOS's stable channel sees major releases every May and November and it has around an extra month of support to give users time to upgrade. Between these releases, only updates that fall into the category of bug fixes or security updates are seen.

  A rolling release version, Nix Unstable, does exist.

- **Alpine Linux**

  A minimal low-footprint beast. Alpine follows the same release schedule as NixOS. However, the main repository is supported for 2 years.
  It doesn't look like you can skip releases like in Fedora, so it's quite the upgrade path if you decide to wait the full 2 years.

  Alpine is very minimal by design. The base system installation has a very small footprint, the core userspace is chosen with minimalism in mind (OpenRC, Busybox, Musl, etc) and the binaries are compiled with special flags for size optimization.

  Alpine is a very secure system, and a great choice for servers, as well as embedded servers with its wide CPU architecture support. Using Alpine for the desktop, however, requires a bit of patience, as the package repository isn't the largest and there isn't as much focus to provide a full set of desktop applications.

- **Ubuntu LTS**

  The king of Linux. The LTS variant of Ubuntu follows a 2 year release cycle, always in April.
  Its approach is the same as most LTS distros, doing a freeze of everything but security and bug fixes.

  Every Ubuntu LTS release stays supported for 5 years and users can extend support further if they use an Ubuntu Pro subscription, extending the lifetime from 5 to 10 years.
  Ubuntu is slowly but surely adopting their home made Snap packaging as a way to give recent software to their long term distros, allowing most users to stay on their LTS releases without the age of the software getting in the way.
  Expect to see an immutable version of Ubuntu that uses only Snaps very soon.

  Lots of distros are derivatives from Ubuntu LTS, including ones like Linux Mint, Pop!_OS, Zorin OS, etc.
  Keep in mind that these distros might take a different approach at when and how users should upgrade, don't follow Ubuntu's LTS upgrading path strictly in those cases. Like with Debian, listen to your derivative first!

  But do expect that it will allow you at least 2 years of support, no matter the derivative based on an LTS, unless they treat it like a regular Ubuntu release.
  Derivatives like to quickly jump to the new LTS as soon as it's released.

Conclusion

After knowing what differentiates distros, you can choose your own as long as you know what's different in each. In case you don't know much about the current distros out there, here's some indications and recommendations:

Stable release systems

- Debian, most Debian derivatives, Fedora, OpenSUSE Leap, NixOS, Alpine Linux, Linux Mint, Ubuntu, most Ubuntu derivatives,

Rolling release systems

- Arch Linux, NixOS unstable, Debian testing and unstable, Void Linux, Artix Linux, OpenSUSE Tumbleweed, Manjaro Linux, EndeavourOS

Systems that don't use SystemD

- Gentoo (optionally), Void Linux, Artix Linux, Devuan, Alpine Linux

Systems with a manual installation process

- Arch Linux (optional), Gentoo, NixOS (optional), Chimera Linux

Server-friendly systems

- Debian, RHEL, Alma Linux, Rocky Linux, Fedora Server, Alpine Linux, NixOS (stable)

Beginner-friendly systems

- Linux Mint, Ubuntu, Fedora, Pop!_OS, KDE Neon, Spiral Linux

Intermediate-friendly systems

- EndeavourOS, Debian, OpenSUSE, Fedora, Spiral Linux, PeppermintOS, Arch Linux (with archinstall)

Advanced systems

- Arch Linux, Void Linux, Alpine Linux, NixOS, Gentoo

There's a lot to choose so, to narrow down the possibilities, here are my personal favorites for different mindsets, in no particular order:

New to Linux:

- Linux Mint
- Fedora

Simplicity and convenience:

- Linux Mint
- Fedora

Power and freedom:

- Arch Linux
- EndeavourOS
- Gentoo
- Void Linux
- Debian
- NixOS

There can't be a concrete answer, as each person has different preferences and necessities. Hopefully this article helped you to choose what Linux system to use. If you feel like a different system than yours interests you or probably suits you better, give it a try on a virtual machine first, and then make a choice on whether to switch or not.